Ανάλυση Κοινωνικών Δικτύων (Social Network Analysis)

Εντοπισμός Κοινοτήτων (Community Detection)

Συμεών Παπαβασιλείου (papavass@mail.ntua.gr)
Βασίλειος Καρυώτης (vassilis@netmode.ntua.gr)

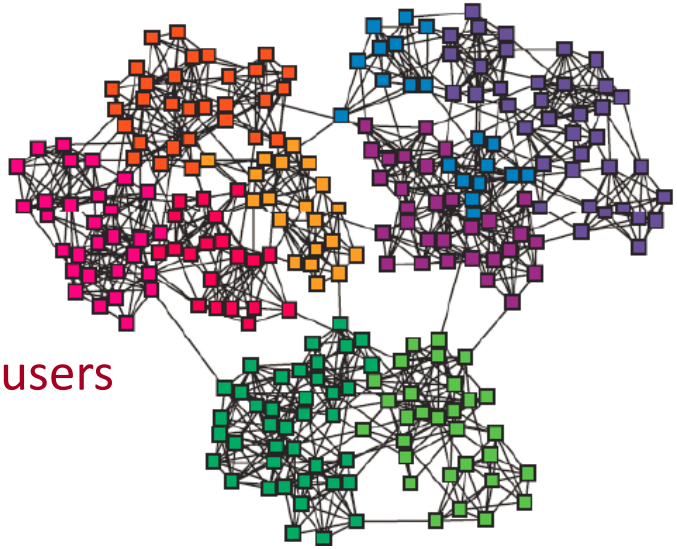
Topics

- Community Definition & Examples
- Community Detection Methods
  - Node-centric
  - Group-centric
  - Network-centric
  - Hierarchy-centric
- Evaluation of Community Detection Methods

What is a Community?

- **Community:** formed by individuals such that those within a group interact with each other more frequently than with those outside the group
  - Other terms: group, cluster, cohesive subgroup, module
- **Community detection:** discovering groups in a network where individuals' group memberships are not explicitly given

- **Explicit**:
  - the result of conscious human decision
- **Implicit**:
  - emerging from the interactions & activities of users
  - need special methods to be discovered



Examples

1. Customers with similar interests or geographic location could be clustered to help recommendation systems

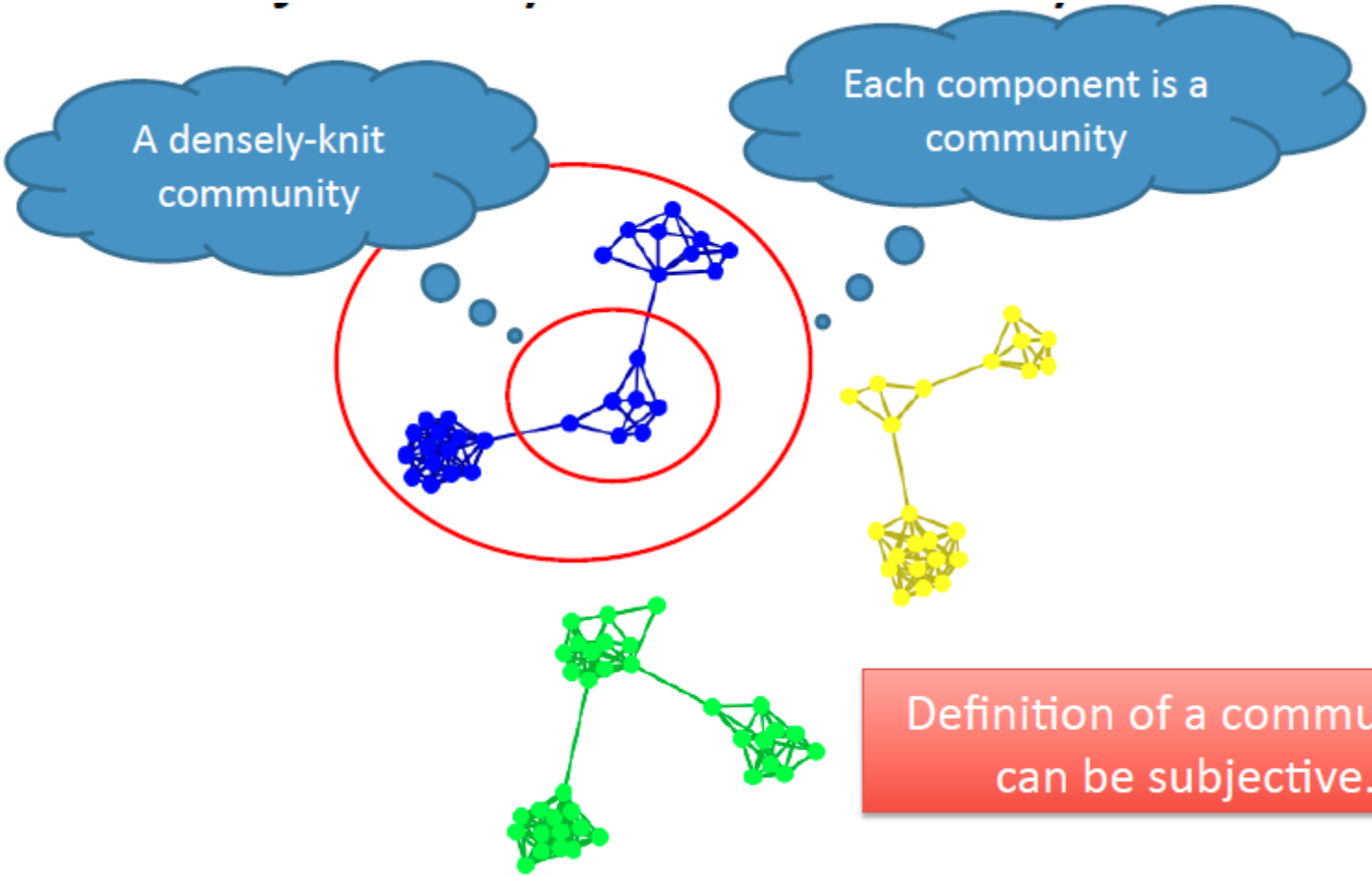
2. Clusters in large graphs can be used to create data structures for efficient storage of graph data to handle queries or path searches

3. Study the relationship among nodes

4. Hierarchical organization study

5. WWW: pages and hyperlinks
   - Identification of clusters can improve page ranking

Subjectivity of Community Definition



Taxonomy of Community Detection Methods

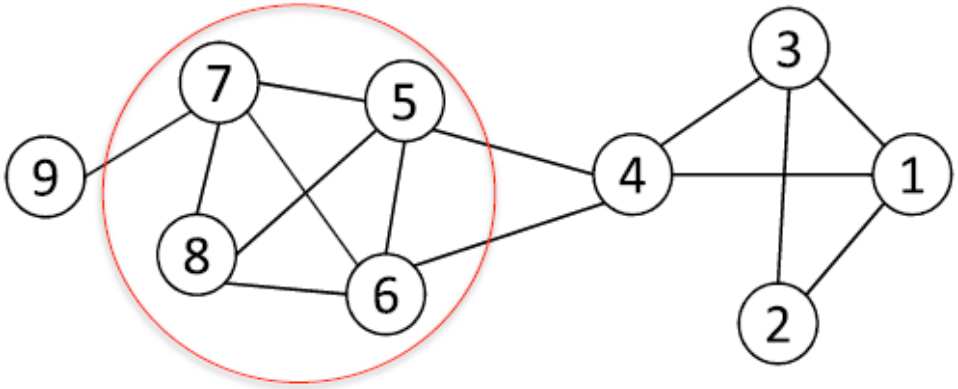
- Node-Centric Community
  - Each node in a group satisfies certain properties
- Group-Centric Community
  - Consider the connections within a group as a whole
  - The group has to satisfy certain properties without zooming into node-level
- Network-Centric Community
  - Partition the whole network into several disjoint sets
- Hierarchy-Community
  - Construct a hierarchical structure of communities

Node-Centric Methods

- Clique Based

Clique Based Method (1)

- **Clique:** a maximum complete subgraph in which all nodes are adjacent to each other
- NP-hard
  - Straightforward implementation to find cliques ➡ very expensive in complexity

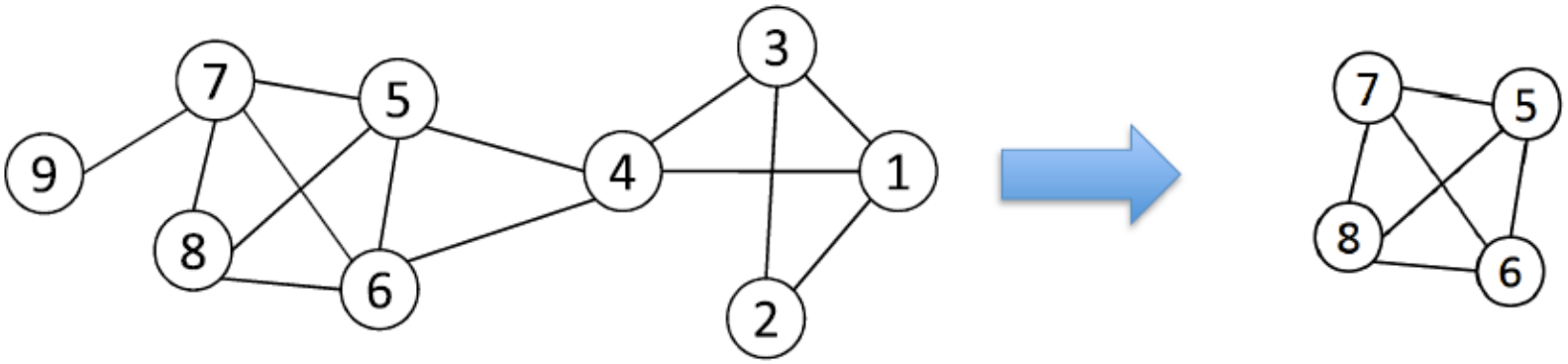


Nodes 5, 6, 7 and 8 form a clique

Clique Based Method (2)

- In a clique of size k, each node maintains degree >= k-1
- Nodes with degree < k-1 will not be included in the maximum clique
- Recursively apply the following pruning procedure
  - Sample a sub-network from the given network, and find a clique in the sub-network, e.g., by a greedy approach **(lower bound of clique size)**
  - Suppose the clique above is size k, in order to find out a *larger* clique, all nodes with degree <= k-1 should be removed
- Repeat until the network is small enough
- In social networks many nodes will be pruned due to their power law degree distribution (scale-free)
- Not stable: usually employed as core for subsequent expansion for a community

Clique Based Method (3)



- Suppose we sample a sub-network with nodes {1-5} and find a clique {1, 2, 3} of size 3
- In order to find a clique >3, remove all nodes with degree <=3-1=2
  - Remove nodes 2 and 9
  - Remove nodes 1 and 3
  - Remove node 4

Group-Centric Methods

(density-based)

Density-Based Groups

- The group-centric method requires the group as a whole to satisfy a certain condition
  - e.g., the group density >= a given threshold

A subgraph $G_s(V_s, E_s)$ is a $\gamma - dense$ quasi-clique if

$$\frac{|E_s|}{|V_s|(|V_s|-1)/2} \geq \gamma$$

Threshold determining the lowest network density

A similar strategy to that of cliques can be used

- Sample a subgraph, and find a maximal $\gamma - dense$ quasi-clique (say, of size k)
- Remove nodes with degree $< k\gamma$

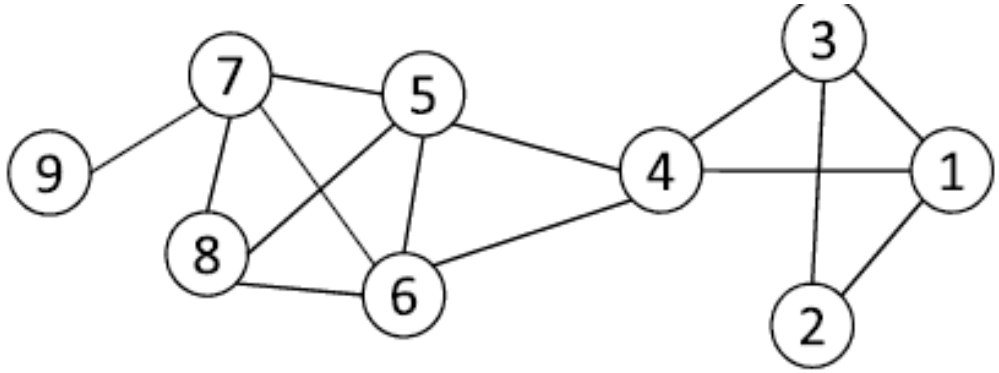
Network-Centric Methods

- Clustering based on vertex similarity
- Spectral clustering
- Modularity maximization

Clustering based on Similarity (1)

- Vertex similarity is defined in terms of the similarity of their neighborhood
- Structural equivalence: two nodes are structurally equivalent iff they are connecting to the same set of nodes ⟹ Too restricted for practical application
- Same community ≡ Same equivalence class

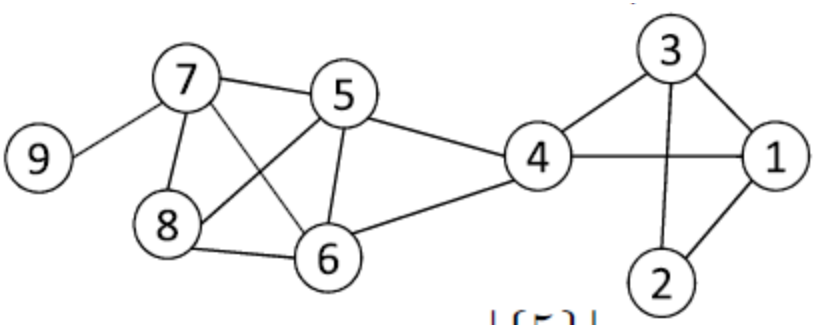
Nodes 1 and 3 are structurally equivalent; So are nodes 5 and 7.



Clustering based on Similarity (2)

Vertex Similarity

- Jaccard Similarity $Jaccard(v_i, v_j) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|} = \frac{\sum_k A_{ik} A_{jk}}{|N_i| + |N_j| - \sum_k A_{ik} A_{jk}},$
- Cosine similarity $Cosine(v_i, v_j) = \frac{\sum_k A_{ik} A_{jk}}{\sqrt{\sum_s A_{is}^2 \cdot \sum_t A_{jt}^2}} = \frac{|N_i \cap N_j|}{\sqrt{|N_i| \cdot |N_j|}}.$



$$Jaccard(4,6) = \frac{|\{5\}|}{|\{1,3,4,5,6,7,8\}|} = \frac{1}{7}$$

$$cosine(4,6) = \frac{1}{\sqrt{4 \cdot 4}} = \frac{1}{4}$$

Determine the similarity measure and then apply e.g., k-means (next slide)

K-means clustering (1)

$x^{(i)}$ may correspond to the nodes' coordinates

Input:

- K (number of clusters)
- Training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$ $x^{(i)} \in \mathbb{R}^n$

K-means optimization objective

$c^{(i)}$ = index of cluster (1,2,…,K) to which example $x^{(i)}$ is currently assigned

μ_k = cluster centroid k ($\mu_k \in \mathbb{R}^n$)

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned

Optimization objective:

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

$$\min_{\substack{c^{(1)},\ldots,c^{(m)},\\ \mu_1,\ldots,\mu_K}} J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$$

K-means clustering (2)

K-means algorithm

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

Repeat {

for i = 1 to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

for k = 1 to K

μ_k := average (mean) of points assigned to cluster k

}

Spectral Clustering (1)

Approximation of minimum radio cut and normalized cut:

$$\widetilde{L} = \begin{cases} D - A & \text{graph Laplacian} \\ I - D^{-1/2} A D^{-1/2} & \text{normalized graph Laplacian} \end{cases}$$

$$D = diag(d_1, d_2, \cdots, d_n)$$ A diagonal matrix of degrees

***K*-communities:** *k* eigenvectors corresponding to k smallest (or largest) eigenvalues and then k-means clustering

Big Data analytics: used for dimensionality reduction

Spectral Clustering (2)

Simple adjacency matrix can be also used

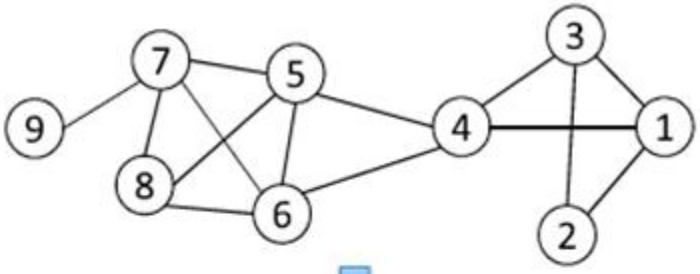
Given a set of points $S = \{s_1, \ldots, s_n\}$ in $\mathbb{R}^l$ that we want to cluster into k subsets:

1. Form the affinity matrix $A \in \mathbb{R}^{n \times n}$ defined by $A_{ij} = \exp(-\|s_i - s_j\|^2/2\sigma^2)$ if $i \neq j$, and $A_{ii} = 0$.
2. Define D to be the diagonal matrix whose (i,i)-element is the sum of A's i-th row, and construct the matrix $L = D^{-1/2} A D^{-1/2}$.[1]
3. Find $x_1, x_2, \ldots, x_k$, the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix $X = [x_1 x_2 \ldots x_k] \in \mathbb{R}^{n \times k}$ by stacking the eigenvectors in columns.
4. Form the matrix Y from X by renormalizing each of X's rows to have unit length (i.e. $Y_{ij} = X_{ij}/(\sum_j X_{ij}^2)^{1/2}$).
5. Treating each row of Y as a point in $\mathbb{R}^k$, cluster them into k clusters via K-means or any other algorithm (that attempts to minimize distortion).
6. Finally, assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

Other partitioning schemes can be also used

Spectral Clustering (3)

Example



Two communities:
{1, 2, 3, 4} and {5, 6, 7, 8, 9}

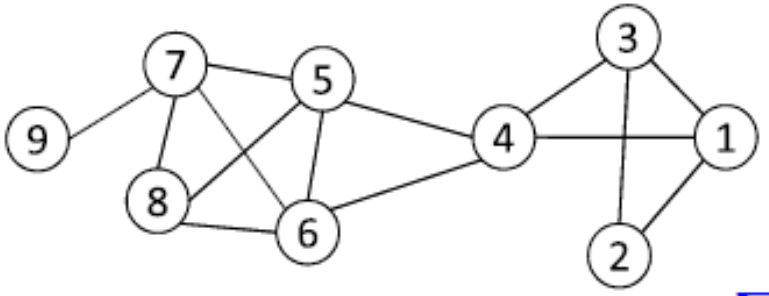
k-means

$$D = diag(3, 2, 3, 4, 4, 4, 4, 3, 1)$$

$$\tilde{L} = D - A = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 4 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 4 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 4 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 4 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix} \Rightarrow S = \begin{bmatrix} 0.33 & -0.38 \\ 0.33 & -0.48 \\ 0.33 & -0.38 \\ 0.33 & -0.12 \\ 0.33 & 0.16 \\ 0.33 & 0.16 \\ 0.33 & 0.30 \\ 0.33 & 0.24 \\ 0.33 & 0.51 \end{bmatrix}$$

Modularity (1)

- Modularity measures the strength of a community partition by taking into account the degree distribution
- Given a network with m edges, the expected number of edges between two nodes with d_i and d_j is $d_i d_j / 2m$



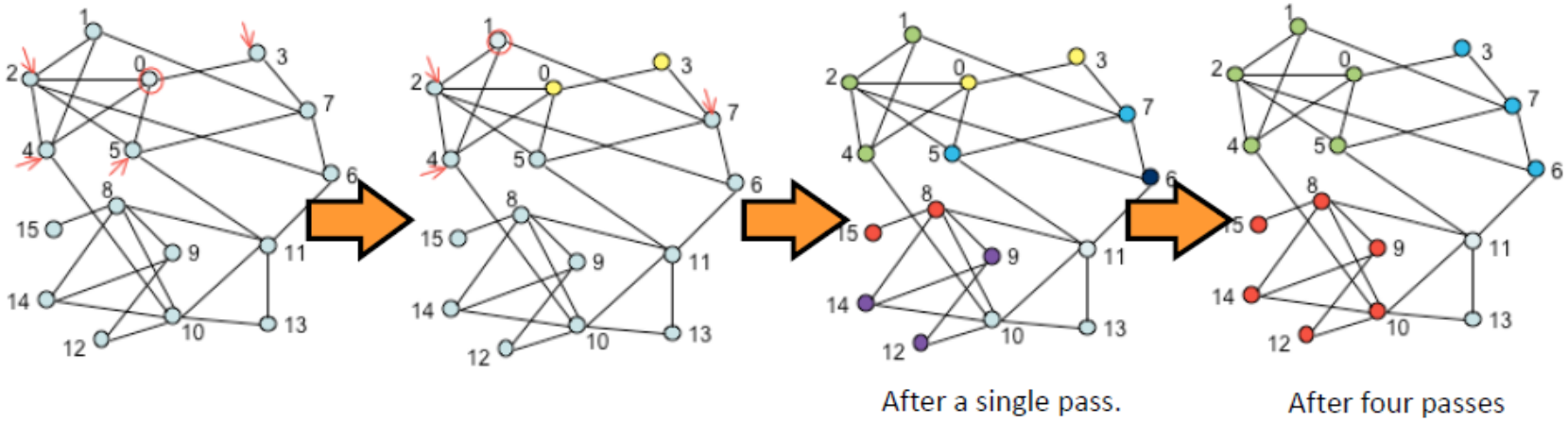
The expected number of edges between nodes 1 and 2 is 3*2/ (2*14) = 3/14

- Strength of a community: $\sum_{i \in C, j \in C} A_{ij} - d_i d_j / 2m$
- Modularity: $Q = \frac{1}{2m} \sum_{\ell=1}^{k} \sum_{i \in C_\ell, j \in C_\ell} (A_{ij} - d_i d_j / 2m)$
- A larger value indicates a good community structure

Modularity (2)

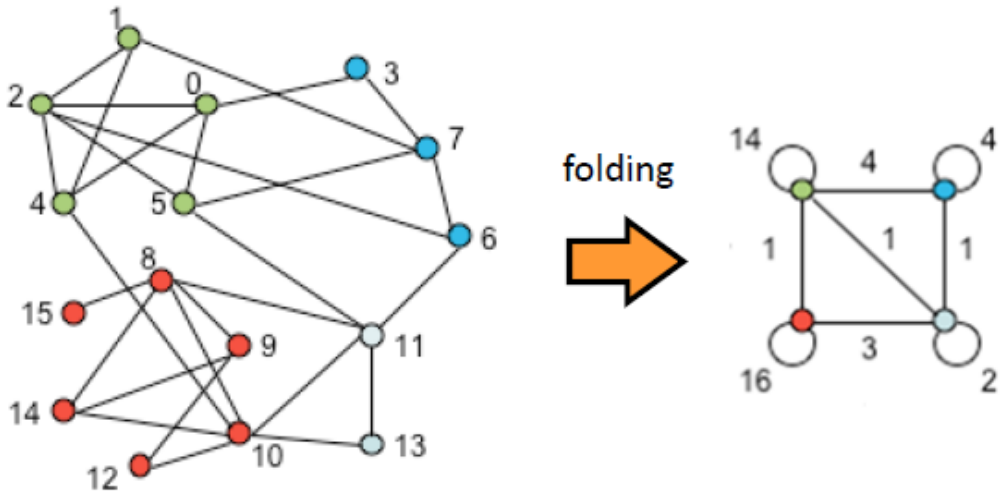
- Modularity ranges from -1 to 1.
- It is positive if the number of edges inside the group are more than the expected number.
  - Variation from 0 indicate difference with random case.
- Finding the configuration with maximum modularity in a graph ⇒ NP complete problem.
- However there are good approximation algorithms.

Modularity (3)



- Initially, each node belongs to its own community (N nodes ⟹ N communities)
- We go through each node with a standard order. To each node, we assign the community of their neighbor as long as this leads to an increase in modularity
- This step is repeated many times until a local modularity maximum is found

Modularity (4)



- **Folding:** Create new graph in which nodes correspond to the communities detected in the previous step
- Edge weights between community nodes are defined by the number of inter-community edges
- Folding ensures rapid decrease in the number of nodes that need to be examined and thus enables large-scale application of the method

Hierarchy-Centric Methods

- Divisive Hierarchical Clustering (Girvan-Newman)
- Agglomerative Hierarchical Clustering

Hierarchical Clustering

- Goal: build a hierarchical structure of communities based on network topology
- Allow the analysis of a network at different resolutions

- **Representative approaches:**
  - Divisive Hierarchical Clustering

Divisive Hierarchical Clustering

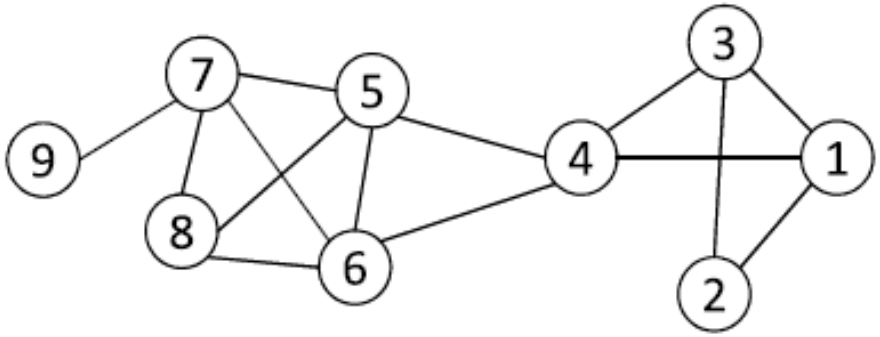
- Partition nodes into several sets
- Each set is further divided into smaller ones
- Network-centric methods can be applied for the partition

- One particular example: ***recursively remove the "weakest" tie***
  - Find the edge with the least strength
  - Remove the edge and update the corresponding strength of each edge
  - Recursively apply the above two steps until a network is decomposed into the desired number of connected components
  - Each component forms a community

Edge Betweenness (1)

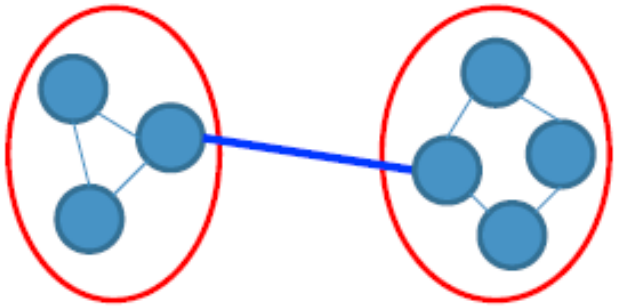
- The strength of a tie can be measured by edge betweenness
- Edge betweenness: the number of shortest paths that pass along with the edge

$$\text{edge-betweenness(e)} = \Sigma_{s<t} \frac{\sigma_{st}(e)}{\sigma_{s,t}}$$

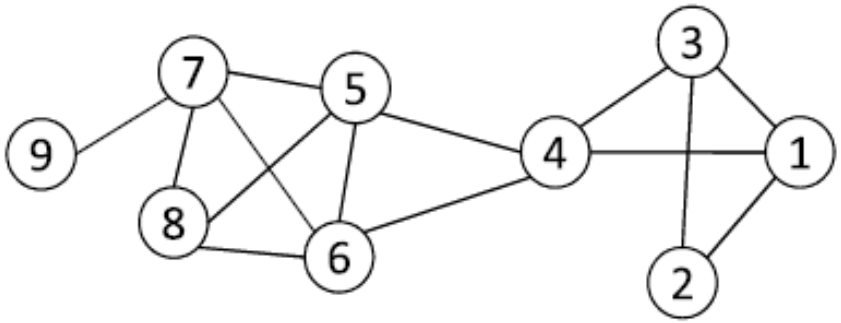


The edge betweenness of e(1, 2) is 4, as all the shortest paths from 2 to {4, 5, 6, 7, 8, 9} have to either pass e(1, 2) or e(2, 3), and e(1,2) is the shortest path between 1 and 2

- The edge with higher betweenness tends to be the bridge between two communities.



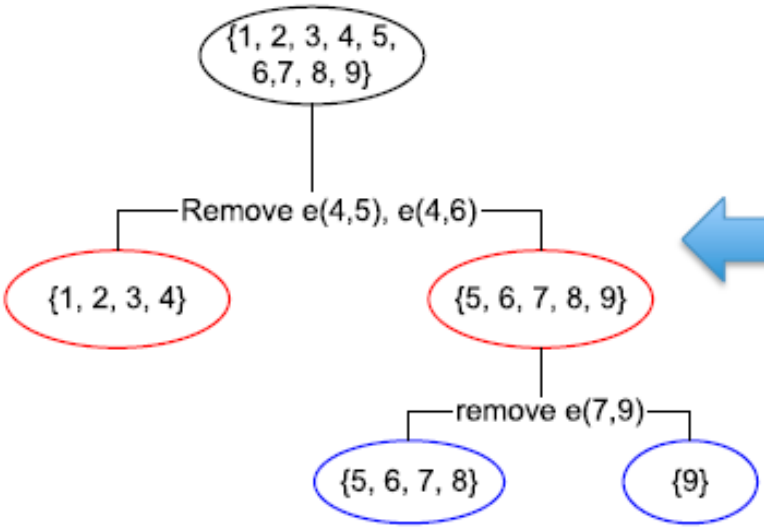
Edge Betweenness (2)



Initial betweenness value

Table 3.3: Edge Betweenness

	1	2	3	4	5	6	7	8	9
1	0	4	1	9	0	0	0	0	0
2	4	0	4	0	0	0	0	0	0
3	1	4	0	9	0	0	0	0	0
4	9	0	9	0	10	10	0	0	0
5	0	0	0	10	0	1	6	3	0
6	0	0	0	10	1	0	6	3	0
7	0	0	0	0	6	6	0	2	8
8	0	0	0	0	3	3	2	0	0
9	0	0	0	0	0	0	8	0	0



After remove e(4,5), the betweenness of e(4, 6) becomes 20, which is the highest;

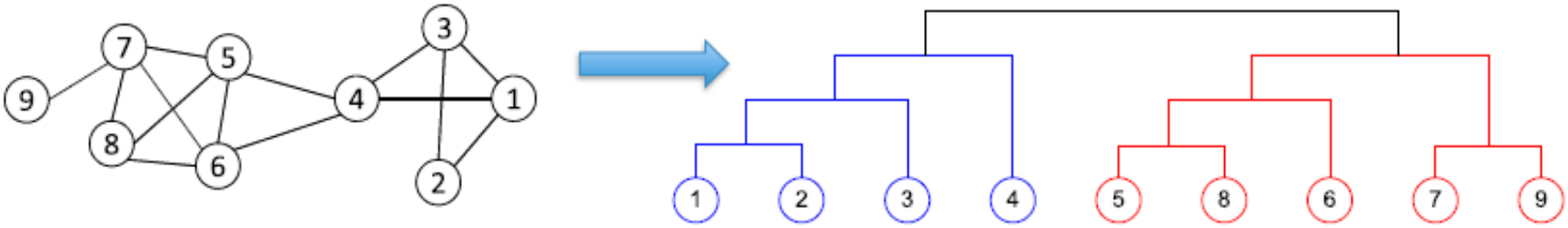
After remove e(4,6), the edge e(7,9) has the highest betweenness value 4, and should be removed.

Stopping Criterion: Modularity Value

Agglomerative Hierarchical Clustering

- Initialize each node as a community

- Merge communities successively into larger communities following a certain criterion
  - E.g., based on modularity increase



Evaluation of Community Detection

- Many methods of community detection due to lack of ground truth information about a community structure in a real-world network
- Chosen method depends on application

➢ **Evaluation Methods:**

- For groups with clear definitions
  - E.g., Cliques, k-cliques, k-clubs, quasi--cliques

  Verify whether extracted communities satisfy the definition
- For networks with ground truth information
  - Visualization for small number of communities
  - Normalized mutual information for larger number of communities

Networks without Ground Truth

- This is the most common situation
- An option for evaluation is the **cross-validation**

- Extract communities from a (training) network
- Evaluate the quality of the community structure on a network constructed from a different date or based on a related type of interaction

- Evaluate and compare different methods based e.g. on modularity